**INSTITUTO POLITÉCNICO DE BEJA **

**Escola Superior de Tecnologia e Gestão**

**Licenciatura em Engenharia Informática**

**Projeto de Engenharia de Software Parte 1**

Fernando Simões Nº - 19922

Vasco Gomes Nº - 19921

2020 - 2021

**INSTITUTO POLITÉCNICO DE BEJA**

**Escola Superior de Tecnologia e Gestão**

**Licenciatura em Engenharia Informática**

**Projeto de Engenharia de Software Parte 1**

Elaborado por:

Fernando Simões Nº - 19922

Vasco Gomes Nº - 19921

Orientado por:

Engenheira Isabel Brito

Relatório de Engenharia de Software apresentado na

Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Beja

2020-2021

**Resumo**

O que é engenharia de software

O que é um software

Este projeto tem como base simular as diferentes fases do ciclo de desenvolvimento de um software, neste caso, um software que oferece uma solução de comunicação e acompanhamento das crianças, assim como privacidade dos dados das crianças assim como dos seus progenitores, e ainda uma forma de comunicação das creches e infantários com os pais de cada criança. Uma vez desenvolvido um software que dá apoio às instituições que trabalham com crianças estes softwares podem possuir métodos de faturação desenvolvidos em parceria com outras empresas.

**Palavras-chave:** Metodogia Kanban, Software, Infantários, Creches,

Índice

[1. Introdução 3](#_Toc67677026)

[2. Descrição de Software Semelhante 4](#_Toc67677027)

[2.1. ChildDiary 4](#_Toc67677028)

[2.1.1. Rotinas Diárias – Registo de atividade das crianças, assim como refeições, presenças, atividades, horas de sono; 4](#_Toc67677029)

[2.1.2. Planeamento Curricular – O *software* auxilia a planear e avaliar as atividades curriculares; 4](#_Toc67677030)

[2.1.3. Tempo Real – Funcionalidade que permite notificar os pais a qualquer hora do dia, sobre o bem-estar da criança; 4](#_Toc67677031)

[2.1.4. Privacidade – Cada pai apenas tem acesso às informações do seu filho, assim como cada educador tem acesso às informações de uma turma apenas. 4](#_Toc67677032)

[2.2. SoftGold 4](#_Toc67677033)

[2.2.1. Atividades da Vida Diária – Permite á semelhança do *software* anterior, fazer monotorização das atividades diárias das crianças, com acréscimo de monotorização do pessoal docente, uma vez que é uma solução física e digital; 4](#_Toc67677034)

[2.2.2. Gestão de Assiduidade – Esta funcionalidade permite controlar a assiduidade das crianças bem como dos colaboradores; 4](#_Toc67677035)

[2.2.3. Escalas de Serviços dos Recursos Humanos – Os recursos humanos asseguram-se de elaborar horários com os respetivos turnos com atividades para as crianças e os funcionários; 4](#_Toc67677036)

[2.2.4. Gestão de Acessos – Plataforma que permite controlar acessos físicos assim como digitais; 4](#_Toc67677037)

[2.2.5. Serviço de Transporte de Crianças – O *Software* permite gerir de uma forma versátil o transporte das crianças até aos seus lares; 4](#_Toc67677038)

[2.2.6. Soluções de Saúde – O *SoftGold*, estabeleceu uma pareceria com a empresa *MobilWave*, de modo a ser possível disponibilizar as soluções de saúde as crianças assim como aos funcionários. 4](#_Toc67677039)

[2.3. Educa4You 5](#_Toc67677040)

[2.3.1. Caderneta Digital – Plataforma que permite consultar os dados relacionados com as crianças assim como entrar em comunicação de pais para educadores; 5](#_Toc67677041)

[2.3.2. Direção Técnica – Gestão de creche ou jardim de infância, a partir dos *smartphone*, *tablet*, ou computador, assim como acompanhamento da equipa pedagógica, e os trabalhos desenvolvidos pelos mesmos com as crianças; 5](#_Toc67677042)

[2.3.3. Controlo de Presenças – Funcionalidade, que monitoriza o controlo de presenças físicas na instituição dos auxiliares de educação. 5](#_Toc67677043)

[3. Plano de Trabalho a Seguir 6](#_Toc67677044)

[4. Ferramentas CASE 7](#_Toc67677045)

[4.1. Análise: 8](#_Toc67677046)

[4.1.1. Recolha de Informação 8](#_Toc67677047)

[4.1.2. Análise de Informação 8](#_Toc67677048)

[4.1.3. Elaboração de Diagramas 9](#_Toc67677049)

[4.2. Desenho 9](#_Toc67677050)

[4.3. Gestão 9](#_Toc67677051)

[5. Conclusões e Perspetivas de Trabalho Futuro 10](#_Toc67677052)

**Lista de Figuras**

Figura 2.2 - Organograma da Empresa XX **Erro! Marcador não definido.**

Figura 3.1 – xxx **Erro! Marcador não definido.**

**Lista de Tabelas**

Tabela 3.1 – Tabela de … **Erro! Marcador não definido.**

# Introdução

Descrever aqui sucintamente em que consistiu o seu projeto de trabalho. Local de realização, atividades previstas e respetivo planeamento e faseamento.

O último parágrafo deste capítulo deve referir a estrutura do relatório, indicando o que se aborda em cada um dos capítulos seguinte.

**Metodologia Kanban**

Nesta simulação de projeto foi escolhida modelo *Kanban* de entre os vários modelos que a metodologia ágil tem, dá aos elementos da equipa uma visão panorâmica do projeto, é de baixo custo, o que para uma empresa start-up, ou que esteja por exemplo na incubadora de empresas do IPBeja, é vantajoso, assim como otimiza tempo dividindo as tarefas por diferentes elementos do grupo.

Ao trabalhar com este método á semelhança modelo *Scrum*, este também possui um elevado nível de priorização das funcionalidades que aplicação deverá ser constituída, mas ao contrário do *Scrum*, em que o modelo consiste em reuniões diárias, este assenta num estímulo visual de modo a saber minimamente em que fase cada elemento se encontra das suas tarefas, isto tanto pode ser feito utilizando uma plataforma digital como o *JIRA, Trello*, ou na sua forma mais básica um quadro físico á moda do *CSI*.

NOTAS GERAIS

1. Todas as figuras, tabelas, gráficos devem ter numeração sequencial, com o primeiro dígito igual ao número de capítulo, p.e. “Figura 2.1 - Exemplo de Figura”. Neste template isso já está feito
2. A bibliografia deve seguir a norma igual à que tem final deste documento. Todas as referências têm de ser citadas no texto, por exemplo: “… este trabalho segue a metodologia indicada em [1], …”
3. O início de qualquer secção tem de ter sempre algum texto descritivo do que se pretende com essa secção.
4. Todas as imagens, tabelas, etc, têm de estar sempre referidas no texto

# Descrição de Software Semelhante

## Descrição do software do enunciado Growappy

A Growappy o software no qual o nosso trabalho está baseado, uma vez que se encontra no enunciado do projeto, este tem a utilidade de facilitar a comunicação entre os encarregados de educação e os educadores, bem como a organização de informação á cerca das crianças em questão. Este pedaço de software português conta com as seguintes características:

**Reconhecimento Facial** – Este software tem a funcionalidade de enviar fotografias das crianças somente para os encarregados de educação com a ajuda de um algoritmo de reconhecimento facial. Este software tem ainda mais uma funcionalidade que permite ofuscar a face das crianças que se encontram em fotografias de grupo, e permitindo deste modo apenas os encarregados de educação ver as caras dos seus filhos.

**Histórias do dia** – Os educadores podem partilhar o dia a dia dos seus filhos, mantendo os pais atualizados acerca dos mesmos.

**Informações da Criança** – Cada criança tem um portfólio online específico, onde os pais e educadores podem realizar alterações sempre que necessárias, este permite ainda registar a aprendizagem das crianças.

**Notificações** – Os encarregados de educação recebem notificações á cerca dos seu educandos, mantendo sempre a privacidade dos mesmos, uma vez que os seu números não estão visíveis no seu perfil.

**Faturação** – Este é um sistema automático de faturação que permite fazer a visualização de históricos de pagamentos, bem como efetuar pagamentos seguros através de entidade e referência.

**Aulas Remotas** – E por fim como podia de ser devido aos tempos que vivemos, tem sido prioritário existir o ensino a distância, e todas as outras atividades que não exijam, pessoas fisicamente a realizá-las, temos ainda reuniões de pais de forma remota e em direto com a vantagem de as pessoas não deixarem o conforto dos seus lares

No âmbito do software descrito no enunciado deste projeto *Growappy*, encontramos 3 *softwares* semelhantes ao mesmo, sendo estes:

## ChildDiary

Este software é bastante similar ao *Growappy*, que foca bastante na comunicação entre educares e pais, assim como na privacidade das crianças e conteúdo partilhado com os pais.

Algumas das funcionalidades do mesmo são:

### **Rotinas Diárias** – Registo de atividade das crianças, assim como refeições, presenças, atividades, horas de sono;

### **Planeamento Curricular** – O *software* auxilia a planear e avaliar as atividades curriculares;

### Tempo Real – Funcionalidade que permite notificar os pais a qualquer hora do dia, sobre o bem-estar da criança;

### Privacidade – Cada pai apenas tem acesso às informações do seu filho, assim como cada educador tem acesso às informações de uma turma apenas.

## SoftGold

O *software SoftGold*, é mais uma vez bastante parecido aos outros 2 softwares referidos anteriormente, este conta com as funcionalidades:

### Atividades da Vida Diária – Permite á semelhança do *software* anterior, fazer monotorização das atividades diárias das crianças, com acréscimo de monotorização do pessoal docente, uma vez que é uma solução física e digital;

### Gestão de Assiduidade – Esta funcionalidade permite controlar a assiduidade das crianças bem como dos colaboradores;

### Escalas de Serviços dos Recursos Humanos – Os recursos humanos asseguram-se de elaborar horários com os respetivos turnos com atividades para as crianças e os funcionários;

### Gestão de Acessos – Plataforma que permite controlar acessos físicos assim como digitais;

### Serviço de Transporte de Crianças – O *Software* permite gerir de uma forma versátil o transporte das crianças até aos seus lares;

### Soluções de Saúde – O *SoftGold*, estabeleceu uma pareceria com a empresa *MobilWave*, de modo a ser possível disponibilizar as soluções de saúde as crianças assim como aos funcionários.

Esta oferta inclui acesso a receitas eletrónicas, exames complementares de diagnóstico, e ainda uma bases de dados vocacionada para os processos médicos dos utentes.

## Educa4You

Por último, mas não o menos importantes, mas ainda assim menos vocacionada na temática do projeto que é comunicação educador – pai, e acompanhamento das crianças assim como a sua privacidade.

Após alguma pesquisa, este *software* também possui funcionalidades para comunicação entre os encarregados de educação e os educadores de infância.

Na lista seguinte, podemos verificar algumas funcionalidades do *software**Educa4You*:

### Caderneta Digital – Plataforma que permite consultar os dados relacionados com as crianças assim como entrar em comunicação de pais para educadores;

### Direção Técnica – Gestão de creche ou jardim de infância, a partir dos *smartphone*, *tablet*, ou computador, assim como acompanhamento da equipa pedagógica, e os trabalhos desenvolvidos pelos mesmos com as crianças;

### Controlo de Presenças – Funcionalidade, que monitoriza o controlo de presenças físicas na instituição dos auxiliares de educação.

# Plano de Trabalho a Seguir

Nesta secção iremos descrever o plano de trabalho que vamos seguir, e neste caso encontra-se no enunciado do projeto.

**Análise:**

Na fase de análise é nos pedido para efetuar uma recolha de informação, também conhecida como estudo de mercado, este caso específico consiste na realização de entrevistas, assim como pesquisas de sistema semelhantes e por sua vez utilização das *UserStories* dos utilizadores de aplicações similares, para melhoramento do nosso *software*.

De seguida com os dados recolhidos e através de ferramentas como o *Microsoft**Office* e os *MindMaps*, que foram lecionados pela professora desta unidade curricular, é feita uma análise da informação de modo a saber que requisitos o *software* deverá conter.

Por fim com todos os dados e a fase de análise dos mesmos concluída, é elaborado um diagrama de uso assim como de especificação do *software*.

Pensando em 2 programadores numa empresa a fazer o desenvolvimento do projeto, todo este trabalho ira demorar cerca de 1 semana para ficar concluído, uma vez que teríamos de obter aprovação do cliente.

**Desenho:**

Na secção do desenho, são elaborados diagramas *UML*, que são compostos por diagramas estruturais, assim como comportamentais.

O *UML*, também conhecido por *Unified**Modeling**Language*. É uma linguagem gráfica com o objetivo de:

* Especificar
* Visualizar
* Construir
* Documentar

Tudo isto para todas as fases do desenvolvimento de um *software*.

Como foi referido no ponto acima, a duração do desenvolvimento do mesmo estará a volta de 2 semanas, uma vez que haverá sempre coisas por melhorar, de modo a conseguir obter a melhor performance, assim como segurança.

**Gestão:**

A secção de gestão é a última, mas é nesta fase que todo o *software* referente ao projeto vai ser desenvolvido, através das últimas 2 secções são implementadas todas as features referidas pelo cliente, utilizando os ambientes de programação, o controlo de versões e assim como os mecanismos de validação e por fim no final de cada feature desenvolvida realizar o teste para a mesma, á semelhança da apresentação feita pelos programadores do *Grupo**Rolear*, mais concretamente a empresa *Rolear**On*.

Para finalizar esta secção, falta indicar o tempo de trabalho, mas uma vez que nunca tivemos trabalho a sério na área, embora os projetos das mais variadas cadeiras desta licenciatura tenham o objetivo de simular projetos reais com aplicações reais, não conseguimos dizer em concreto quanto tempo demoraria, por esse mesmo motivo fizemos uma estimativa de acordo com a quantidade de features a adicionar, assim como a complexidade das mesmas, uma vez que uma features pode demorar entre 1 e 3 semanas a ser desenvolvida, tendo encontra o que foi acabado de descrever estimamos um tempo de 1 mês e meio a 2 meses e meio de desenvolvimento com poucas *features*.

# Ferramentas CASE

As ferramentas *CASE* (*Computer**Aided**Software**Engeneering*), é um nome dado a todas a ferramentas utilizadas para auxiliar nas várias etapas de desenvolvimento de *software*.

Nas várias etapas do desenvolvimento de um *software*, existem 2 categorias principais, sendo estas *Front**End*, também denominado de *Upper**Case*, em que podemos encontrar as ferramentas que iremos utilizar na primeiras fases do projeto assim como a análise e a gestão.

Por outro lado, temos ainda uma outra categoria denominada de *Back**End*, também conhecida por *Lower**Case*, que auxiliam em fases como a programação, seguida dos testes e por fim, mas não o menos importante a manutenção do *software* ou aplicação, que nesta cadeira é um tema bastante frisado pela professora, uma vez que o *software* construído deve procurar ter a maior longevidade possível.

Para além destas 2 existe ainda uma terceira também conhecida com *Integranted**Case*, em que são *softwares* utilizados todo o percurso de desenvolvimento de um *software*.

Após termos feito uma breve introdução às ferramentas *CASE*, e ficando agora melhor a compreender a temática, vamos agora passar á enumeração e explicação das mesmas.

## Análise:

A secção de análise consiste na Recolha da Informação á cerca do projeto, posterirormente é realizada uma análise da mesma, e por fim é elaborado um diagrama.

### Recolha de Informação

* Google
* Zoom
* Gmail
* MindMap
* Microsoft Word
* Growappy
* ChildDiary
* SoftGold
* Educa4You

A recolha da informação consiste na realização de entrevistas, presencialmente, ou através da plataforma *Zoom*, e o *Gmail*, assim como pesquisas de sistema semelhantes, utilizando o *Google* e por sua vez utilização das *UserStories* dos utilizadores de aplicações similares, para melhoramento do nosso *software,* que se encontram descritas nos últimos 4 pontos.

Utilização de *MindMaps*, para estruturação de informação, assim como a representação com o máximo de detalhes possíveis do *software* em questão.

### Análise de Informação

* Growappy
* Word
* ChildDiary
* SoftGold
* Educa4You

De seguida na secção da análise de informação, analisamos a informação que recolhemos, neste caso das plataformas semelhantes, e utilizamos o *Microsoft**Word* par realizar as anotações do trabalho.

### Elaboração de Diagramas

* VisualParadigm

Para terminar a secção da análise, temos a Elaboração dos diagramas, em que exemplificamos os casos de uso da aplicação.

## Desenho

* VisualParadigm

Na secção do desenho, utilizamos a ferramenta *VisualParadigm*, para realizar modelos e diagramas referentes ao *software*.

## Gestão

* Android Studio
* Visual Studio Code
* JDK
* JUnit
* GitHub
* SQL Server

Por último, na secção da gestão, utilizamos variadas ferramentas, cada qual com o seu propósito, como por exemplo o *Android**Studio* para desenvolvimento de aplicações *Android*, agregado como *JDK* que é o compilador de *Java*, *Visual**Studio**Code*, para fazer todo o desenvolvimento *Web*, quer *Front****-****End*, quer *Back****-****End****,*** de seguida utilização da ferramenta *JUni*t, que é uma framework para testes em Java, o *GitHub*, como controlo de versões, e por fim o *SQL**Server*para o desenvolvimento de uma base de dados, assim como a sua manutenção conforme visto na cadeira de Bases de Dados 2.

# Funcionalidades do software a Desenvolver - JoyfuLearning

Nesta secção iremos descrever todos os requisitos funcionais e não funcionais que o nosso software deverá ter.

## Requisitos Funcionais –

Na nossa opinião, os requisitos funcionais são representativos das funcionalidades do software requeridas pelo cliente.

Ex: Diário da Criança no software Growappy.

Agora vamos dar a conhecer os requisitos funcionais do nosso software – JoyfuLearning:

Comunicação:

* Feed de Notícias;
* Serviço de Mensagens Instantâneas (DM’s);

O requisito da comunicação, é um dos requisitos mais importantes, pois é o objetivo principal do software, que como o próprio nome indica permite existir uma linha direta de comunicação entre os encarregados de educação e os educadores, quer seja utilizando o Feed de Notícias, ou o serviço de mensagens instantâneas.

Caderneta de Registos JoyfuLearning:

* Assiduidade da Criança;
* Saúde da Criança;
* Lista de Atividades Diárias;
* Atividades Realizadas;
* Alimentação da Criança;

De seguida encontramos a caderneta de registos, esta funcionalidade permite guardar um historial sobre as várias atividades e informações da criança, como a assiduidade, a saúde, atividades diárias, atividades realizadas, e alimentação da criança.

Aprendizagem Remota:

* Repositório de Trabalhos;
* Ferramenta de Gravação de Aulas;

Par finalizar, mas não menos importante nos dias de hoje, temos ainda a funcionalidade que permite realizar aulas remotas para salvaguardar as crianças em caso de situação epidemiológica e também em caso de necessidade de permanecer em casa, sem ter a ver com o Covid 19, assim como uma ferramenta para realizar a gravação de aulas, quer sejam presenciais, quer sejam online.

## Requisitos não funcionais –

Por outro lado, os requisitos não funcionais entendemos por funcionalidades que não estão explícitos ao cliente, mas são necessárias para o bom funcionamento da aplicação.

Ex: Algoritmo de Reconhecimento Facial no software Growappy.

Após a elaboração dos requisitos funcionais, temos os requisitos não funcionais:

* Privacidade dos Dados;
* Eficiência do Software nas tarefas;
* Facilidade de Uso;
* Traduções;
* Paradigma de Programação Orientava por Objetos (Java);
* Requisitos Legais;

Uma vez descritos os requisitos funcionais, iremos agora passar aos requisitos não funcionais, e explicá-los devidamente, começamos pela privacidade dos dados, uma vez que se os educadores e encarregados de educação não quisessem privacidade plataformas como estas não existiam, utilizavam o Facebook, segue-se a eficiência do software a executar tarefas, uma vez que ninguém nos dias de hoje tem tempo de esperar que o software demore a carregar as funcionalidades, assim como o software tem de ser fácil de usar, caso contrário os utilizadores não usam e torna-se um flop.

Seguimos para as traduções em que por exemplo pode haver alunos cujos pais tem diferentes nacionalidades, e preferem nem que seja o inglês, já do modo de vista de todas as aplicações desenvolvidas hoje em dia, tudo usa o paradigma da programação orientada por objetos, uma vez que é mais fácil de fazer alterações ás estruturas de dados e procura diminuir o gap semântico entre a vida real e a programação.

Para finalizar, temos os recursos legais, pois caso contrário levas com um processo em cima.

# Conclusões e Perspetivas de Trabalho Futuro

Conclusões do trabalho desenvolvido, com referência comparativa também com os objetivos iniciais (que constam do primeiro capítulo).

A seção de “Perspetivas de Trabalho Futuro” deve indicar o que pode ser desenvolvido futuramente por si, quer no âmbito das atividades realizadas, quer no que este estágio contribuiu para o seu desenvolvimento profissional e o que pode ser o seu percurso futuro.

**Bibliografia**

[1] F. P. J. Brooks, The Mythical Man-Month Essays on Software Engineering Anniversary Edition. Addison-Wesley, 1995.

[2] P. P.-S. Chen, \The entity-relationship model | toward a uni\_ed view of data," ACM Trans. Database Syst., vol. 1, n. 1, pp. 9{36, Mar\_co 1976. [Online]. Disponível: http://doi.acm.org/10.1145/320434.320440

[3] (2013) Android developers: Developer tools. [Online]. Disponível: http://developer.android.com/tools/index.html

[4] J. Hu, H. Wu, A. Jennings, e X. Wang, “Fast and robust equalization: A case study," in Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, (SCI 2000), Florida, USA, 23-26 July 2000. FL, USA: International Institute of Informatics and Systemics, 2000, pp. 398.

<Outras referências>